IN THE CLAIMS

Please amend the claims as follows.

1. (Previously Presented) A method of maintaining a cache memory in a computer system having a first processor, a first memory, and at least a first cache between the first processor and the first memory, the method comprising:

performing a first memory access to the first memory by the first processor;

storing a first cache line from the first memory into the first cache;

performing an instruction that enables a cache-invalidate function to be performed by the first processor upon execution of a resource-synchronization instruction;

performing, within the first processor, a resource-synchronization instruction that operates on a first memory location in the first memory, wherein performing the resource synchronization instruction includes modifying the first memory location in the first memory and invalidating at least the first cache line in the first cache during execution of the resource synchronization instruction on the first memory location in the first memory;

performing a second memory access to the first memory by the first processor;

storing a second cache line from the first memory into the first cache;

performing an instruction that disables the cache-invalidate function from being performed by the first processor upon execution of the resource-synchronization instruction;

performing the resource-synchronization instruction on a second memory location in the first memory, wherein performing the resource synchronization instruction includes modifying the second memory location in the first memory without invalidating the second cache line in the first cache.

2. (Previously Presented) The method of claim 1, wherein the computer system further includes a second processor and at least a second cache between the second processor and the first memory, the method further comprising:

performing a third memory access to the first memory by the second processor;

storing a third cache line from the first memory into the second cache;

performing the instruction that enables the cache-invalidate function to be performed by the second processor upon execution of the resource-synchronization instruction;

performing, within the second processor, a resource-synchronization instruction that operates on a third memory location in the first memory, wherein performing the resource synchronization instruction includes modifying the third memory location in the first memory and invalidating at least the third cache line in the second cache during execution of the resource synchronization instruction on the third memory location in the first memory, and wherein no data in the first cache is invalidated;

performing a fourth memory access to the first memory by the second processor;

storing a fourth cache line from the first memory into the second cache;

performing the instruction that disables the cache-invalidate function from being performed by the second processor upon execution of the resource-synchronization instruction;

performing, within the second processor, a resource-synchronization instruction that operates on a fourth memory location in the first memory, wherein performing the resource synchronization instruction includes modifying the fourth memory location in the first memory, wherein at least the fourth cache line is not invalidated in the second cache.

3. (Original) The method of claim 2, wherein the resource-synchronization instruction is a test-and-set instruction.

4. (Original) The method of claim 3, wherein the instruction that enables the cache-invalidate function is an enable-test-and-set-invalidate instruction, and the instruction that disables the cache-invalidate function is an disable-test-and-set-invalidate instruction.

5. (Original) The method of claim 1, wherein the resource-synchronization instruction is a test-and-set instruction.

6. (Original) The method of claim 5, wherein the instruction that enables the cache-invalidate function is an enable-test-and-set-invalidate instruction, and the instruction that disables the cache-invalidate function is an disable-test-and-set-invalidate instruction.

7. (Previously Presented) The method of claim 1, wherein invalidating at least the first cache line in the first cache during execution of the resource synchronization instruction on the first memory location in the first memory includes invalidating the entire first cache.

8-26. (Cancelled)